

Universidad Autónoma de Madrid

Escuela Politécnica Superior



Doble Grado en Ingeniería Informática y Matemáticas

TRABAJO DE FIN DE GRADO

GOOGLE PLAY GAME SERVICES

Carolina Gutiérrez Suárez
Tutor: Alejandro Sierra Urrecho

Julio 2016

GOOGLE PLAY GAME SERVICES

Autor: Carolina Gutiérrez Suárez
Tutor: Alejandro Sierra Urrecho

Departamento de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid

Julio 2016

Agradecimientos

En primer lugar quiero agradecer a mi tutor Alejandro por apoyarme a pesar de todas las dificultades.

A todos mis compañeros y amigos por los buenos momentos y la ayuda prestada.

A Juan por estar siempre ahí y darme ánimos.

Y sobre todo, a mi familia. Especialmente a mis padres y a mi hermana por confiar en mí y estar conmigo a pesar de la distancia.

Resumen

Resumen

Año tras año las aplicaciones móviles han demostrado estar en auge. El número de aplicaciones crece de forma exponencial y la finalidad de las mismas es muy diversa. Desde aplicaciones dedicadas al entretenimiento (música, libros, películas...), a la fotografía, a los negocios, etc. Entre todas estas categorías, destacan claramente las aplicaciones de juegos.

Google ofrece multitud de servicios tales como Google Maps, Google+ o Google Analytics. Algunos de estos servicios representan una oportunidad para mejorar las aplicaciones Android permitiendo, por ejemplo, hacer un seguimiento del uso de las mismas, añadir mapas o mensajes desde la nube. Sin embargo, los servicios de Google no forman parte de la plataforma Android sino que es necesario instalarlos en cada dispositivo de forma individual. Entre estos servicios se encuentra Google Play Games Services.

Google Play Games Services es un servicio de Google que facilita el desarrollo de juegos tanto para móviles como para ordenador (versión web). En el caso de aplicaciones móviles para Android, Google Play Games Services ofrece la posibilidad de incluir en los juegos logros y tablas de clasificación, eventos, modo multijugador en tiempo real, modo multijugador por turnos, estadísticas, almacenamiento de partidas, etc.

El objetivo de este proyecto es entender cómo funciona dicho servicio, comprobar su funcionalidad y limitaciones e investigar sobre las ventajas y desventajas del mismo. Como resultado se pretende elaborar una aplicación sencilla, un juego, para Android que utilice el servicio Google Play Games, con el objetivo de publicar posteriormente el resultado final en la Play Store.

Palabras Clave

Android - Google Play Games Services - Google Play - Play Store

Abstract

Year over year mobile applications have shown to be rising in popularity. The number of applications is increasing exponentially and their purpose is varies from one to another. From applications whose theme is entertainment (music, books, movies ...), to photography, to business, etc. Among all these categories, game applications clearly stand out.

Google offers many services such as Google Maps, Google+ or Google Analytics. Some of these services provide an unique opportunity to improve Android applications allowing, for instance, to monitor the use of them, add maps or enable messaging from the cloud. However, Google services are not part of the Android platform by default, so they need to be installed on each device individually. One of these services is Google Play Games Services.

Google Play Games Services is a Google service that simplifies the development of games for mobile devices and desktop (web version). In the case of mobile applications for Android, Google Play Games Services offers the possibility of including achievements and leaderboards, events, real-time multiplayer mode, turn-based multiplayer mode, statistics, saved games, etc. in the games with ease. The aim of this project is to understand how this service works, research its functionality and limitations and investigate the advantages and disadvantages of its use. As a result, it is intended to develop a simple application (game) for Android that uses the Google Play Games service, in order to subsequently publish the final result in the Play Store.

Key words

Android - Google Play Games Services - Google Play - Play Store

Índice general

1. Introducción	1
1.1. Motivación del proyecto	1
1.2. Objetivos	2
1.3. Estructura del documento	2
2. Estado del arte	5
2.1. Google Play Games Services	5
2.1.1. Ventajas e inconvenientes	8
2.2. Consola de desarrollador	10
2.3. Servidor DADM	10
3. Definición del proyecto	13
3.1. Alcance	13
3.2. Metodología	13
3.3. Requisitos	14
3.3.1. Requisitos funcionales	14
3.3.2. Requisitos no funcionales	15
3.4. Herramientas utilizadas	15
4. Diseño	17
4.1. Selección del juego	17
4.2. Descripción del juego y sus normas	18
4.3. Diseño de la aplicación	19
4.4. Interfaz gráfica	19
5. Desarrollo	21
5.1. Codificación	21
5.1.1. Estructura del proyecto	21
5.1.2. Modo multijugador basado en turnos	23
5.2. Compilación y pruebas	25
6. Pruebas y Resultados	27
6.1. Pruebas	27
6.2. Resultados	28

7. Conclusiones y trabajo futuro	31
Glosario	33
Referencias	35
Anexos	37
A. Manual de usuario	39
A.1. Instalación de la aplicación	39
A.2. Uso de la aplicación	40
B. Manual del desarrollador	43
B.1. Registro en la consola de desarrollador de Google Play	43
B.2. Configuración del juego en la consola de desarrollador	44
B.2.1. Vincular aplicaciones	45
B.3. Publicación del juego en la Play Store de Google	47
B.3.1. Firma del APK para la publicación	49

Índice de tablas

5.1. Documentación del método <code>getSelectOpponentsIntent()</code>	23
5.2. Documentación del método <code>createMatch()</code>	23
5.3. Documentación del método <code>takeTurn()</code>	24
5.4. Documentación del método <code>finishMatch()</code>	24
5.5. Documentación del método <code>rematch()</code>	24

Índice de figuras

2.1. Ventaja de las actualizaciones de Google Play Games	9
4.1. Icono de la aplicación	17
4.2. Tablero inicial de Reversi	18
5.1. Estructura del proyecto	22
5.2. Ejemplo de mensaje del servidor	25
5.3. Firma APK en modo debug	25
6.1. Menú principal de la aplicación	28
6.2. Pantalla de juego de la aplicación	28
6.3. Valoración de la aplicación por los usuarios	29
A.1. Pantalla de la Play Store	40
A.2. Pantalla de inicio en la aplicación por primera vez	40
A.3. Pantalla de configuración de los datos de usuario	41
A.4. Menú de inicio de la aplicación	41
A.5. Pantalla de selección de oponente	42
B.1. Consola de desarrollador de Google Play	44
B.2. Pantalla principal de la consola de desarrollador	44
B.3. Información requerida para la publicación del juego	45
B.4. Menú de selección para el tipo de aplicación	45
B.5. Vinculación una aplicación de Android	46
B.6. Creación de cliente de OAuth	47
B.7. Fichero ids.xml del proyecto	47
B.8. Pantalla inicio de publicación del juego	48
B.9. Clasificación de contenido de la publicación	48
B.10. Pantalla de éxito en la publicación	49
B.11. Firma APK en modo release	50

1

Introducción

Este Trabajo de Fin de Grado tiene como propósito la familiarización con el servicio de Play Games de Google. Para ello se ha llevado a cabo el desarrollo de una aplicación Android que utiliza dicho servicio. En los siguientes apartados se explican la motivación y los objetivos principales del trabajo y se describe la estructura del resto de la memoria.

1.1. Motivación del proyecto

Uno de los muchos servicios que ofrece Google es el servicio de juegos Google Play. Éste permite integrar fácilmente, tanto en juegos para móviles como web, características comunes a diversos juegos, tales como logros, clasificaciones, almacenar partidas, modo multijugador en tiempo real, modo multijugador por turnos, estadísticas de juegos (para aplicaciones Android), entre otras. Este servicio de Google Play está disponible para el desarrollo de aplicaciones tanto Android como iOS.

Además, al ser un servicio de Google Play, tiene la ventaja de que el desarrollador puede contar con las APIs más recientes de forma simple. En el caso de las aplicaciones Android todas las actualizaciones de los servicios de Google Play se distribuyen de forma automática a través de la Play Store y las últimas versiones de la librería del cliente se actualizan a través del SDK Manager de Android.

1.2. Objetivos

El objetivo principal de este Trabajo de Fin de Grado es llevar a cabo una prueba de concepto centrada técnicamente en el empleo de Google Play Games Services: entender cómo funciona y trasladar este conocimiento al desarrollo de una aplicación de prueba que muestre su funcionamiento. Además, se publicará el resultado obtenido en la Play Store de Google.

Este objetivo viene motivado por la posibilidad de utilizar dichos servicios en la asignatura optativa Desarrollo de Aplicaciones para Dispositivos Móviles del cuarto curso de Ingeniería Informática y quinto curso del Doble Grado en Ingeniería Informática y Matemáticas. En dicha asignatura los alumnos deben desarrollar una aplicación Android. El principal problema es que, una vez terminado el curso, los alumnos no pueden publicar sus proyectos porque sus aplicaciones dependen de un servidor interno de la Universidad. Además, la utilización de los servicios estándar de Google podría simplificar significativamente el desarrollo y permitir realizar aplicaciones más completas con mayor funcionalidad. Por último, la documentación de estos servicios (tanto en forma de manuales como foros de preguntas/respuestas) facilitaría el desarrollo de la aplicación por parte de los alumnos, en el contexto de la asignatura.

Para cumplir con los objetivos anteriormente mencionados se decide desarrollar un juego sencillo que utilice Google Play Games Services, en concreto el juego multijugador por turnos Reversi.

1.3. Estructura del documento

Este trabajo se divide en siete capítulos que forman el contenido principal, pero además cuenta con un Glosario, Referencias y dos Anexos. A continuación se describe el propósito de cada uno de ellos.

El primer capítulo sirve de presentación del proyecto, en él se explican la motivación y los objetivos del trabajo.

En el capítulo 2 se realiza un análisis del estado del arte.

En el capítulo 3 se define el alcance de la aplicación que se va a desarrollar, la metodología seguida y los requisitos de la aplicación así como las herramientas utilizadas para el desarrollo del proyecto.

En el capítulo 4 se describe el diseño de la aplicación.

En el capítulo 5 se documenta la implementación de la aplicación desarrollada.

En el capítulo 6 se explican los resultados obtenidos.

En el capítulo 7 se exponen las conclusiones finales sobre el trabajo realizado.

Seguidamente se presenta un glosario con la terminología empleada y un apartado con las referencias utilizadas. Por último, se encuentran los anexos.

En el anexo A se incluye un manual de usuario en el que se explica brevemente cómo utilizar la aplicación.

En el anexo B se incluye un manual de desarrollador en el que se describen los pasos necesarios para publicación del juego en la Play Store, detallando el registro en la consola de desarrollador, la configuración de la aplicación y finalmente la publicación del juego.

2

Estado del arte

2.1. Google Play Games Services

Google Play Games Services permite crear juegos más sociales, con logros, tablas de clasificación, juegos guardados en la nube, juegos multijugador en tiempo real, etc. Están disponibles para Android, iOS y cualquier otro dispositivo conectado [1].

En el caso de desarrollo de juegos para Android las principales características que ofrece este servicio son:

- **Logros (Achievements)**

Incluir logros al juego puede ser una manera de aumentar la participación de los usuarios dentro del juego y recompensar al jugador por determinadas acciones. Además, permite a los jugadores comparar su progreso con el de otros. Se pueden incluir 3 tipos de logros: logro oculto (los detalles sobre el logro se ocultan al jugador), logro revelado (significa que el jugador sabe de la existencia del logro pero no lo ha conseguido todavía) y logro desbloqueado (el jugador ha conseguido con éxito el logro) [2].

- **Tablas de clasificación o ranking (Leaderboards)**

Las tablas de clasificación pueden ser una forma de aumentar la competencia entre los jugadores. El servicio de Google Play se hace cargo de la gestión de la mayoría de los aspectos de esta tabla de clasificación, comprueba si la puntuación de un jugador es mejor que la entrada de tabla de clasificación actual del jugador para la puntuación diaria, semanal o de todos los tiempos. Además, los juegos pueden tener múltiples tablas de clasificación, por ejemplo, una por cada nivel. Existen dos

versiones diferentes de cada tabla de clasificación para el jugador: la clasificación social donde la clasificación está compuesta por personas de los círculos del usuario; o la tabla de clasificación pública, que es una tabla de clasificación formada por jugadores que han elegido compartir su actividad de juego públicamente [3].

■ **Modo multijugador en tiempo real (Real-time Multiplayer)**

La API de multijugador en tiempo real [4] permite conectar varios jugadores juntos en una sola sesión de juego (partida) y el envío de mensajes entre los jugadores. Dicha API se encarga de gestionar las conexiones de red para crear y mantener un ambiente multijugador en tiempo real. También proporciona una interfaz de usuario de selección de jugadores para invitar a los jugadores a unirse a la partida, dichos jugadores pueden ser seleccionados específicamente, emparejados de forma aleatoria o bien una combinación de ambos, hasta alcanzar un máximo de 8 jugadores. Este servicio almacena información de los participantes y el estado de la partida durante el ciclo de vida del juego multijugador. Además permite enviar las invitaciones y actualizaciones del juego a los jugadores.

Las invitaciones son enviadas por los servicios de Google Play a través de las notificaciones de Google Cloud a los dispositivos Android, y a través del servicio Apple Push Notification (APN) para dispositivos iOS.

■ **Modo multijugador por turnos (Turn-based Multiplayer)**

En un juego de multijugador por turnos, los jugadores participan en turnos alternados actualizando los datos del juego durante la partida. Se turnan de manera asíncrona de acuerdo con el orden establecido por el juego y la partida tendrá un único estado compartido que cada jugador podrá modificar en su turno.

La API multijugador por turnos [5] proporcionada por Google Play Games Services almacena información acerca de los jugadores y del estado de la partida en los servidores de Google, compartiendo los datos actualizados de las partidas de forma asíncrona con todos los participantes y mostrando notificaciones a los jugadores. Si el juego se ejecuta en un dispositivo móvil, la API multijugador basada en turnos proporciona además por defecto una interfaz de usuario de selección de jugadores.

Esta API permite enviar invitaciones a amigos, buscar jugadores al azar o una combinación de ambos. Además, en la interfaz de usuario por defecto proporcionada por Google también se ofrece la opción "jugadores cercanos", que permite que jugadores que estén físicamente cerca se unan a la partida. Las partidas pueden tener hasta un máximo de 8 participantes y como son asíncronas no es necesario que todos los jugadores estén conectados al servicio de juegos al mismo tiempo para desarrollarla.

Actualmente, las notificaciones de turno y las invitaciones sólo están disponibles para dispositivos Android e iOS.

■ **Eventos y misiones (Events and Quests)**

Este servicio [6] permite recolectar datos generados por sus jugadores durante las partidas y almacenarlos en los servidores de Google para el análisis del juego. Los eventos permiten realizar un seguimiento de la interacción de los usuarios con

la aplicación. Estos datos se pueden utilizar como retroalimentación sobre cómo mejorar el juego. Por ejemplo, se puede ajustar el nivel de dificultad de ciertos niveles dependiendo de cómo de hábil sea el usuario. Los datos que se quieran recoger se puede definir de forma flexible y se pueden incluir métricas tales como la frecuencia con la que se utiliza un elemento o el tiempo que requiere realizar una acción.

Por otro lado, el servicio de misiones complementa al servicio de eventos permitiendo introducir nuevos retos basados en la información de los eventos. Por ejemplo, las misiones permiten reenganchar o incentivar a los jugadores mediante recompensas.

Tanto los nuevos eventos como misiones en el juego que el desarrollador quiera añadir, son configurados en la consola de Google Play. De este modo no es necesario volver a publicar el juego entero.

Si el juego está instalado en el dispositivo de un jugador, las nuevas misiones se mostrarán automáticamente cuando el jugador inicie la aplicación Google Play Games, sin necesidad de actualización. También se pueden presentar las nuevas misiones a los jugadores a través de la interfaz por defecto proporcionada por Play Games SDK o a través de la propia interfaz de usuario personalizada.

■ **Partidas guardadas (Saved Games)**

El servicio de guardar partidas [7] permite almacenar la progresión dentro del juego de los jugadores en los servidores de Google. Los jugadores pueden continuar una partida desde el punto donde decidieron guardar la partida. Este servicio hace posible sincronizar los datos de una partida del jugador a través de múltiples dispositivos. Por ejemplo, permite empezar una partida en un dispositivo Android y luego seguir jugando en una tableta sin perder ningún progreso.

En cuanto al almacenamiento en la nube de las partidas, a los desarrolladores no se les cobra por los datos guardados del juego. En su lugar, estos datos se deducirán de la cuota de Google Drive del jugador.

Estos datos, como ya se ha mencionado, se almacenan en Google Drive en la carpeta de datos de aplicaciones de sus jugadores. Esta carpeta sólo puede ser leída y escrita por el juego, es decir, no puede ser vista o modificada por los juegos de otros desarrolladores, por lo que hay una protección adicional contra la corrupción de datos. Además, las partidas guardadas están aisladas de la manipulación directa por los jugadores (en un principio).

En cuanto a la compatibilidad sin conexión, el juego puede leer y escribir en una partida guardada. Sin embargo, no será capaz de sincronizar los datos con los servicios de Google Play Games hasta que se establezca la conectividad de red.

Para poder utilizar este servicio es necesario habilitarlo en la consola de desarrollador.

■ **Estadísticas de juego (Player Stats)**

La API de Estadísticas [8] permite crear a medida distintas etapas del ciclo de vida del jugador, es decir, construir experiencias a medida para cada jugador en base a cómo está progresando. Esta API se puede utilizar para tomar acciones proactivas

y fomentar que un jugador menos activo vuelva a participar, por ejemplo, mediante la promoción de nuevos artículos.

Los tipos de datos de los jugadores que se pueden recuperar son: la duración de la sesión promedio del jugador en minutos, el número de días desde la última partida, el número de compras, el percentil de la sesión (que indica el número de sesiones que el actual jugador ha jugado en comparación con el resto de jugadores de este juego), etc.

■ Regalos (Game gifts)

Para hacer el juego más colaborativo y social, se puede permitir a los jugadores enviar y recibir regalos u objetos dentro del juego o solicitarlos mediante el uso de la API de Game Gifts. Google Play Games Services proporciona una interfaz de usuario que hace que sea fácil para los jugadores enviar y solicitar regalos a los amigos.

La API de regalos [9] está disponible actualmente para Android únicamente, a través del SDK de Google Play Services.

■ Conexiones cercanas (Nearby Connections)

La API de Nearby Connections [10] permite descubrir fácilmente otros dispositivos conectados a la misma red local, e intercambiar mensajes en tiempo real. Esta API se puede utilizar como: pizarra colaborativa, invitar a otros usuarios de la red local a unirse al juego en los juegos multijugador local. También permite a los jugadores, en los juegos multi-pantalla, ver una pantalla de juego personalizada en sus dispositivos personales, mientras que todos los participantes pueden ver una pantalla grande compartida, por ejemplo en una Android TV.

2.1.1. Ventajas e inconvenientes

A continuación, se mencionan algunas ventajas e inconvenientes que presenta el uso de Google Play Games Services para el desarrollo de juegos.

Ventajas que ofrece el uso de Google Play Games Services

- Permite desarrollar aplicaciones para distintos sistemas operativos (Android e iOS) y distintos dispositivos (móviles, tablets y ordenadores).
- Proporciona servidores en los que almacenar la información de las partidas.
- Se encarga de la autenticación de usuarios a través de las cuentas de Gmail.
- Proporciona códigos a modo de ejemplo con interfaces de usuario.
- Una vez pagada la cuota de desarrollador no hay que pagar por subir las aplicaciones a la Play Store.

- Las actualizaciones se distribuyen de forma automática (figura 2.1).

Inconvenientes derivados del uso de Google Play Games Services

- Requiere estar en posesión de una cuenta de Google+.
- Para algunos juegos es necesario disponer de conexión a internet.
- Limita el número máximo de jugadores para las partidas a 8.
- Es obligatorio pagar una cuota de desarrollador.
- Es necesario instalar la aplicación *Play Games* para usar una aplicación que utilice el servicio.

Otros

- Utiliza información de tus círculos de amigos de Google+.
- Tiene por defecto un límite de 200.000 consultas al día.
- Los dispositivos Android con versiones anteriores a la 2.3 o los que no posean la aplicación Google Play no tendrán disponible las actualizaciones.

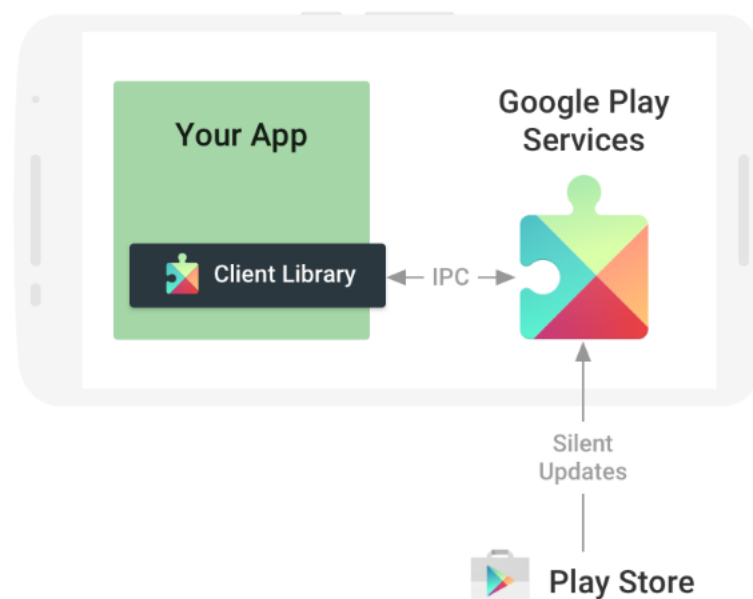


Figura 2.1: Ventaja de las actualizaciones de Google Play Games

2.2. Consola de desarrollador

La consola de desarrollador de Google es una herramienta online proporcionada por Google que permite a los desarrolladores publicar y distribuir aplicaciones [11]. Además, los desarrolladores pueden obtener informes sobre las aplicaciones como la valoración de los usuarios o estadísticas de juego y descargas.

Para desarrollar aplicaciones que hagan uso de los servicios de Google Play Games es necesario:

- Registrarse en la consola de desarrollador y para ello efectuar el pago de una cuota de 25\$.
- Configurar el juego en la consola.

Una vez realizados los pasos anteriores se puede proceder al desarrollo del código de la aplicación. Posteriormente se podrá desde la consola:

- Añadir cuentas de correo electrónico de los usuarios encargados de probar la aplicación.
- Publicar la aplicación en la Play Store de Google.
- Obtener informes y estadísticas sobre el juego.

En el anexo B se puede encontrar un manual detallado de cómo registrarse en la consola, cómo configurar el juego y cómo publicarlo.

2.3. Servidor DADM

El principal objetivo de este Trabajo de Fin de Grado viene motivado por la posibilidad de incluir el sistema de servicios de Google Play Games en la asignatura Desarrollo de Aplicaciones para Dispositivos Móviles.

Aunque la funcionalidad de la aplicación desarrollada en la asignatura y la aplicación desarrollada en este trabajo es algo diferente, las aplicaciones son bastante similares. En ambos casos se desarrolla un juego para Android.

Comparación DADM y Google Play Games Services

- **Bibliotecas**

En ambos casos se hace uso de bibliotecas, en la asignatura se utilizan dos bibliotecas: una proporcionada por los profesores y la biblioteca Volley (para

optimizar el envío de peticiones http desde las aplicaciones Android hacia servidores externos) y en el caso de este trabajo se utiliza una biblioteca que proporciona Google.

- **Servidor**

En la asignatura de móviles se utiliza un servidor de la universidad que aunque cumple con el objetivo requerido, su funcionalidad está limitada mientras que no ocurre así con los servidores de Google.

- **Ficheros de código java**

El número de ficheros de código varía enormemente. Mientras que para la aplicación desarrollada en la asignatura el número de ficheros *.java* se encuentra en torno a los 25-30, la cantidad de ficheros requeridos para desarrollar un juego con los servicios de Google Play Games ha sido de 4.

Para ser justos, tal y como se ha mencionado anteriormente la funcionalidad de las aplicaciones es similar pero no la misma. Por ejemplo, la aplicación desarrollada en la asignatura posee un módulo de estadísticas de juego que no se ha incluido en la aplicación desarrollada para este trabajo.

- **Ficheros de layout**

Para la interfaz de la aplicación Google proporciona interfaces de usuario por defecto por lo que simplifica bastante el desarrollo. De este modo, la aplicación desarrollada para este trabajo cuenta con un único fichero de *layout* mientras que en la asignatura se han requerido en torno a 10 ficheros.

3

Definición del proyecto

En este capítulo se definirán y explicarán el alcance del proyecto, la metodología empleada, los requisitos funcionales y no funcionales del proyecto y las herramientas utilizadas.

3.1. Alcance

Este Trabajo de Fin de Grado tiene como principal objetivo entender cómo funciona Google Play Games Services [1] y crear para ello una aplicación Android que haga uso de dicho servicio. En concreto, se ha optado por realizar un juego multijugador por turnos.

3.2. Metodología

Para el desarrollo de la aplicación se ha optado por seguir un ciclo de vida en cascada. Este modelo ordena de forma rigurosa todas las etapas del proceso de desarrollo software. Una fase no puede iniciarse hasta que se haya terminado por completo la fase anterior.

Se ha elegido este tipo de ciclo de vida debido a la naturaleza del proyecto puesto que requería una primera fase de investigación, posteriormente una puesta en práctica y por último, una muestra del resultado obtenido.

Se muestra a continuación un desglose por tareas:

1. Información

En primer lugar, se ha procedido a la investigación y lectura de documentación a cerca del servicio de Google Play Games.

2. Consola de desarrollador

En segundo lugar, tras haber comprendido el funcionamiento se crea una cuenta de desarrollador para poder hacer uso de dicho servicio.

3. Comprobación de su funcionamiento

En tercer lugar, se ha procedido a la familiarización con el servicio mediante unos códigos de prueba proporcionados por Google. Se ha entendido cómo se pueden implementar las distintas funcionalidades que ofrece su servicio.

4. Definición de la aplicación

Antes de comenzar con el desarrollo de la aplicación se define qué se espera que haga y se definen una serie de requisitos que debe cumplir.

5. Desarrollo de la aplicación

Durante esta fase se realizan de forma simultánea el desarrollo de código y las pruebas.

6. Publicación de la aplicación

Por último, una vez que se ha conseguido una aplicación completamente funcional se procede a la publicación de la misma

3.3. Requisitos

En esta sección se enumeran los requisitos de la aplicación. Estos se dividen en dos clases: requisitos funcionales y requisitos no funcionales. Los primeros hacen referencia al comportamiento esperado de la aplicación mientras que los segundos hacen referencia a los atributos de calidad que se esperan.

3.3.1. Requisitos funcionales

RF. 1 Los usuarios accederán a la aplicación iniciando sesión con su cuenta de Gmail.

RF. 2 Los usuarios podrán crear partidas del juego reversi.

RF. 3 Cada partida tendrá exactamente dos jugadores y los jugadores se turnarán para efectuar los movimientos.

RF. 4 Los usuarios podrán invitar a otros jugadores a unirse a la partida.

RF. 5 Los usuarios podrán seleccionar la opción de *jugador automático* que permite que se produzca un auto-emparejamiento aleatorio. Cuando un usuario selecciona esta

opción puede unirse a una partida existente o quedar a la espera de que se una otro jugador.

RF. 6 Los usuarios podrán editar su imagen y su id.

RF. 7 Los usuarios podrán permitir que otros usuarios los encuentren por su id o por su cuenta de correo electrónico.

RF. 8 Los usuarios podrán permitir que otros usuarios vean su actividad de juegos.

RF. 9 Los usuarios podrán habilitar la opción de *jugadores cercanos* que permite que otros jugadores que se encuentren en la misma zona les inviten a jugar.

RF. 10 Los usuarios podrán consultar la lista de partidas. En la lista se mostrarán las invitaciones a nuevas partidas, las partidas en las que sea su turno, las partidas en las que sea el turno del contrincante y las partidas finalizadas.

RF. 11 Una vez finalizada una partida los usuarios podrán pedir a su rival la revancha.

3.3.2. Requisitos no funcionales

RNF. 1 La aplicación será compatible con dispositivos cuyo sistema operativo sea Android y su versión sea la 2.3 o superior.

RNF. 2 La aplicación requerirá de conexión a internet para la transmisión de los datos.

RNF. 3 El sistema no requerirá paradas programadas para su mantenimiento.

RNF. 4 La aplicación hará uso de la pantalla táctil del dispositivo como método de interacción con el usuario.

RNF. 5 El sistema aportará retroalimentación al usuario de forma que en cada momento sepa qué está ocurriendo y el estado de las acciones que intenta realizar. Dicha retroalimentación será mostrada mediante un *toast* o un *dialog* con un texto claro y conciso sobre el estado de la acción que se está realizando.

3.4. Herramientas utilizadas

A continuación, se especifican las herramientas que han sido necesarias para el desarrollo de este proyecto exponiendo su utilidad. Las necesidades cubiertas por dichas herramientas incluyen:

- Programar la aplicación para el sistema operativo Android.
- Llevar a cabo un control de versiones.

- Probar la aplicación.
- Diseñar el icono de la aplicación.
- Publicar la aplicación.

Entorno para programar la aplicación Android

Android Studio [12] es el IDE (Entorno de desarrollo integrado) oficial para programar aplicaciones Android. Además, es totalmente gratuito y está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux. Otra ventaja que tiene Android Studio es que soporta una gran variedad de sistemas de control de versiones entre los que se incluyen Git (y GitHub), CVS, Mercurial, Subversion, y Google Cloud Source Repositories. Por estas razones se ha elegido Android Studio como entorno de desarrollo de la aplicación.

Control de versiones

Para llevar a cabo un control de versiones del código de la aplicación se ha optado por emplear Git [13] como soporte de control y Bitbucket [14] como plataforma donde alojar el repositorio. Se ha optado por Bitbucket porque además de ser gratuita permite crear repositorios privados.

Pruebas

Para poder probar la aplicación ha sido necesario utilizar la consola de desarrollador de Google y dos dispositivos físicos. En la consola de desarrollador es posible habilitar las cuentas de correo electrónico autorizadas para probar la aplicación.

Diseño del icono de la aplicación

Para diseñar el logo de la aplicación y demás iconos requeridos se ha empleado Libre Office Draw. Se ha elegido dicha herramienta por estar incluida dentro del paquete de software de oficina gratuito y de código abierto LibreOffice [15], y además está diseñada para ser compatible con los principales paquetes ofimáticos.

Publicación de la aplicación

Para publicar la aplicación se ha utilizado la consola de desarrollador de Google [11].

4

Diseño

En este capítulo se expone la selección del juego elegido y se explica brevemente en qué consiste. También se describe el diseño de la aplicación a desarrollar teniendo en cuenta los requisitos que ésta debe cumplir.

4.1. Selección del juego

Para poner a prueba los servicios de Google Play en los que se basa este Trabajo de Fin de Grado se ha decidido implementar un juego para Android. El juego elegido para ello es el juego multijugador basado en turnos Reversi. Se ha elegido este juego porque es relativamente sencillo. La complejidad de desarrollo de la aplicación no se centra en el juego seleccionado sino en los servicios utilizados.

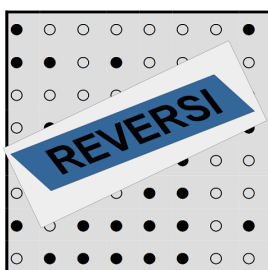


Figura 4.1: Icono de la aplicación

En la figura 4.1 se muestra el logo de la aplicación.

4.2. Descripción del juego y sus normas

Reversi es un juego que enfrenta a dos jugadores sobre un tablero cuadrado con 64 casillas donde el objetivo es acabar la partida con un número de fichas superior al del oponente sobre el tablero.

Los jugadores tendrán fichas de colores diferentes, por ejemplo, blancas y negras. El juego comienza con las cuatro casillas centrales del tablero ocupadas con dos fichas de cada color formando una diagonal entre sí tal y como se muestra en la figura 4.2

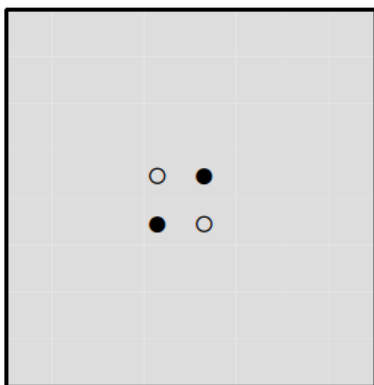


Figura 4.2: Tablero inicial de Reversi

Normas del juego

El primer movimiento lo realizará el jugador que juegue con las fichas negras y posteriormente se irán alternando el turno ambos jugadores.

Los movimientos consisten en incorporar fichas al tablero no en desplazar fichas que ya estuvieran. Sólo se podrá incorporar una ficha en cada turno y cuando ésta esté flanqueando a una o varias fichas contrarias. Es decir, colocar la nueva ficha en un extremo de una hilera de fichas del color del contrario en cuyo extremo opuesto hay una ficha del color del jugador que tiene el turno. Además, no pueden existir casillas libres entre ninguna de ellas. Esta hilera puede ser vertical, horizontal o diagonal. De este modo, las fichas del contrincante quedan encerradas entre una que ya estaba en el tablero y la nueva ficha.

Cada vez que un jugador incorpora una ficha al tablero encerrando a otras del jugador contrario, pasa a hacer suyas las fichas encerradas, es decir, cambian de color.

Si cuando se incorpora una ficha se provoca una situación de flanqueo en más de una línea, se darán la vuelta a todas las fichas contrarias que estuvieran implicadas en cada uno de los flanqueos. Si no fuera posible para un jugador encerrar a ninguna ficha, deberá pasar en su turno, volviendo el mismo a su oponente.

Por último, la partida finaliza cuando todas las casillas del tablero son ocupadas o ninguno de los 2 jugadores tiene posibilidad de incorporar una nueva ficha. Gana el jugador que tiene más fichas sobre el tablero.

4.3. Diseño de la aplicación

Con el objetivo de que el desarrollador se familiarice con la funcionalidad que ofrecen los servicios de Google Play, Google proporciona códigos de ejemplo de juegos.

Para este trabajo se ha tomado como referencia el juego *SkeletonTbmp*. El proyecto Android de este juego tiene únicamente dos ficheros, uno encargado del formato de los datos y otro que se encarga de toda la lógica y la conexión con los servicios de Google Play Games.

Para que la aplicación pueda cumplir con la funcionalidad requerida se decide estructurar el proyecto de la siguiente forma:

- Por un lado, se conservan los ficheros proporcionados y se limita la funcionalidad al formato de datos y la conexión con el servidor.
- Por otro lado, se decide crear un módulo encargado de la lógica del juego en sí. Es decir, del funcionamiento completo de una partida. Para ello se decide utilizar dos ficheros, el primero dibujará el tablero de juego y responderá a los toques de pantalla y el segundo, en función de los movimientos recibidos por la clase anterior comprobará si efectivamente se puede mover y actualizará el tablero.

4.4. Interfaz gráfica

El juego *SkeletonTbmp* también proporciona una interfaz de usuario. De modo que para esta aplicación sólo ha sido necesario diseñar una interfaz para el tablero de juego y adaptar el resto de pantallas.

La interfaz de usuario que proporciona el juego tomado de ejemplo es relativamente simple y consiste en un fichero *layout*. Para diseñar el tablero de juego de la aplicación, dado que su funcionalidad es más compleja, se implementará una clase.

5

Desarrollo

5.1. Codificación

Antes de comenzar a desarrollar un juego que utilice Google Play Games Services es necesario registrarse en la consola de desarrollador [11]. Posteriormente, hay que configurar el juego en la consola (para más información sobre el registro y la configuración en la consola de desarrollador se recomienda consultar el anexo B).

Además, para poder hacer uso de las APIs de Google Play se requiere tener configurado en el proyecto el SDK de los servicios de Google Play y haber añadido en los ficheros del proyecto los ids que se muestran en la consola.

5.1.1. Estructura del proyecto

El proyecto cuenta con 4 ficheros de código java: *GpgsReversiActivity.java*, *GpgsReversiTurn.java*, *TableroReversi.java* y *TableroReversiView.java*, además de la librería proporcionada por Google, *BaseGameUtils* y los ficheros de recursos (figura 5.1).

A continuación se detalla la funcionalidad de cada uno de los ficheros:

- **GpgsReversiActivity.java**

Es la actividad principal de la aplicación, se encarga de conectar todas las clases del proyecto y del envío y recepción de datos. Esta actividad implementa `ApiClient.ConnectionCallbacks`, `ApiClient.OnConnectionFailedListener`, `OnInvitationReceivedListener`, `OnTurnBasedMatchUpdateReceivedListener`, y también `View.OnClickListener`.

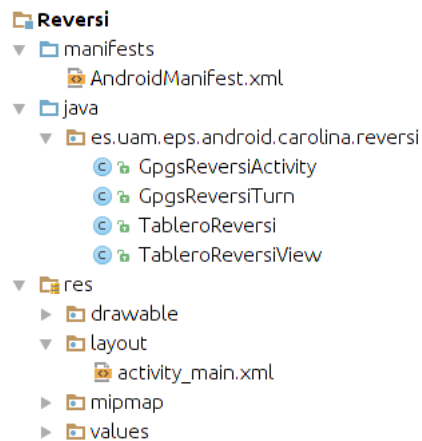


Figura 5.1: Estructura del proyecto

■ GpgsReversiTurn

Esta clase es la encargada del formato de los datos que se envían y reciben del servidor. Posee dos atributos que son el contador de movimientos efectuados y una cadena que representa la codificación del tablero de juego.

■ TableroReversi

Esta clase se encarga de la lógica del juego. Comprueba si los movimientos son válidos, efectúa los movimientos sobre el tablero y codifica el tablero.

■ TableroReversiView

Esta clase extiende de la clase View, y se encarga de la representación del tablero así como de determinar qué fichas han sido seleccionadas.

En cuanto a los ficheros de recursos, se enumeran a continuación las diferentes carpetas empleadas en la aplicación:

- *layout*: contiene un único fichero *.xml* en el que se definen las diferentes pantallas de la aplicación salvo el tablero de juego que es una clase (TableroReversiView).
- *mipmap*: contiene el icono de la aplicación en sus distintas resoluciones (drawable-mdpi, drawable-hdpi, drawable-xdpi y drawable-xxdpi).
- *values*:
 - *colors.xml*: contiene los códigos de los colores que emplea la aplicación.
 - *dimens.xml*: contiene las medidas de los componentes gráficos.
 - *ids.xml*: contiene los ids que se obtienen de la consola de desarrollador y son necesarios para el uso de los servicios de Google Game Play.
 - *strings.xml*: contiene el texto que aparece en la interfaz gráfica.

- *styles.xml*: contiene los estilos que se aplican de manera específica a algunos componentes gráficos.
- *tbmp_helper_strings.xml*: contiene el texto que se muestra en caso de error en la conexión con el servidor de juegos.

5.1.2. Modo multijugador basado en turnos

Para desarrollar un juego multijugador basado en turnos es necesario haber habilitado en la consola de desarrollador dentro de *aplicaciones vinculadas* la opción *Multijugador por turnos*.

A continuación, se comentan los métodos más relevantes relacionados con la implementación del modo multijugador.

■ **public abstract Intent getSelectOpponentsIntent (GoogleApiClient apiClient, int minPlayers, int maxPlayers, boolean allowAutomatch)**

Este método devuelve un intent que permitirá al usuario enviar invitaciones a otros jugadores a una partida. Puesto que Reversi es un juego para dos jugadores (el usuario que envía la invitación y el que la recibe) se le pasa como parámetro mínimo y máximo de jugadores (sin incluir al jugador actual) 1. Además, en la llamada de este método se puede especificar si se desea habilitar la opción de auto-emparejamiento que, para este juego, se ha permitido. En la tabla 5.1 se indican los parámetros que recibe y el retorno de este método.

Nombre	getSelectOpponentsIntent()
Parámetros	apiClient: la clase GoogleApiClient encargada de interactuar con las APIs
	minPlayers: número mínimo de jugadores sin incluir al jugador actual
	maxPlayers: número máximo de jugadores sin incluir al jugador actual
	allowAutomatch: indica si se permite o no la opción de auto-emparejamiento
Retorno	Intent

Tabla 5.1: Documentación del método getSelectOpponentsIntent()

■ **public abstract PendingResult<TurnBasedMultiplayer.InitiateMatchResult> createMatch (GoogleApiClient apiClient, TurnBasedMatchConfig config)**

Este método se encarga de crear una nueva partida. Es posible indicar si el oponente puede ser un jugador asignado por auto-emparejamiento, que para este juego se podrá. En la tabla 5.2 se indican los parámetros que recibe y el retorno de este método.

Nombre	createMatch()
Parámetros	apiClient: la clase GoogleApiClient encargada de interactuar con las APIs
	config: parámetros de configuración para la partida a crear
Retorno	PendingResult: para acceder a los datos cuando estén disponibles.

Tabla 5.2: Documentación del método createMatch()

- **public abstract PendingResult<TurnBasedMultiplayer.UpdateMatchResult> takeTurn(GoogleApiClient apiClient, String matchId, byte[] matchData, String pendingParticipantId)**

Este método permite actualizar los datos de una partida. Se pasa como parámetro el id del jugador que posee el próximo turno. En caso de que no se proporcione el id del siguiente participante se esperará a que se una a la partida un jugador aleatorio de auto-emparejamiento. En la tabla 5.3 se indican los parámetros que recibe y el retorno de este método.

Nombre	takeTurn()
Parámetros	apiClient: la clase GoogleApiClient encargada de interactuar con las APIs
	matchId: id de la partida a actualizar
	matchData: información actualizada de la partida
	pendingParticipantId: id del próximo jugador que posee el turno o null si se está a la espera de un jugador aleatorio
Retorno	PendingResult: para acceder a los datos cuando estén disponibles.

Tabla 5.3: Documentación del método takeTurn()

- **public abstract PendingResult<TurnBasedMultiplayer.UpdateMatchResult> finishMatch(GoogleApiClient apiClient, String matchId)**

Este método sirve para indicar que la partida ha terminado con el reciente movimiento de uno de los jugadores. En la tabla 5.4 se indican los parámetros que recibe y el retorno de este método.

Nombre	finishMatch()
Parámetros	apiClient: la clase GoogleApiClient encargada de interactuar con las APIs
	matchId: id de la partida a actualizar
Retorno	PendingResult: para acceder a los datos cuando estén disponibles.

Tabla 5.4: Documentación del método finishMatch()

- **public abstract PendingResult<TurnBasedMultiplayer.InitiateMatchResult> rematch (GoogleApiClient apiClient, String matchId)**

Este método permite solicitar a cualquiera de los jugadores la revancha de una partida que acaba de terminar. El jugador que la solicite comienza la partida. En la tabla 5.5 se indican los parámetros que recibe y el retorno de este método.

Nombre	rematch()
Parámetros	apiClient: la clase GoogleApiClient encargada de interactuar con las APIs
	matchId: id de la partida que se quiere volver a jugar
Retorno	PendingResult: para acceder a los datos cuando estén disponibles.

Tabla 5.5: Documentación del método rematch()

Envío y recepción de datos

Además de los métodos anteriormente mencionados, en la implementación del modo multijugador basado en turnos es importante el envío y recepción de datos. Es decir, el

cómo los usuarios se turnan y actualizan el tablero de juego. Tal y como se comentaba en la subsección 5.1.1, la clase encargada del formato de envío y recepción de datos es la clase *GpgsReversiTurn*. Esta clase maneja archivos JSON que contienen el contador de la partida y un string de datos que representa el tablero. La información manejada por esta clase es imprescindible para poder llevar a cabo un juego de este tipo.

Sin embargo, la información que se envía al servidor es mucho más completa. Incluye la descripción del juego, información sobre los participantes, nombre del desarrollador, nombre del paquete del proyecto, el estado de la partida, el turno y mucho más (figura 5.2).

```
TurnBasedMatchEntity{Game=GameEntity{ApplicationId=920453897400, DisplayName=ReverseGameTFC, PrimaryCategory=Juegos de mesa,
SecondaryCategory=null, Description=El Reversi es un juego que enfrenta a dos jugadores sobre un tablero cuadrado con 64 casillas
donde el objetivo es acabar la partida con un número de fichas superior al del oponente sobre el tablero..., DeveloperName=CarolinaG5,
IconImageUrl=content://com.google.android.gms.games.background/images/d59ebf4d/180, IconImageUrl=https://lh3.googleusercontent.com/9eVgfnv-
pmHj62sRkoLj_NZaZv5h7ZVEyJUsop0RxoZ4REGWAwY-XHXVvs4mjbJE=s96, HiResImageUrl=content://com.google.android.gms.games.background/images/
d59ebf4d/181, HiResImageUrl=https://lh3.googleusercontent.com/9eVgfnv-pmHj62sRkoLj_NZaZv5h7ZVEyJUsop0RxoZ4REGWAwY-XHXVvs4mjbJE=s282,
FeaturedImageUrl=content://com.google.android.gms.games.background/images/d59ebf4d/182, FeaturedImageUrl=https://lh3.googleusercontent.com/
PMLrP61a1PWRQRKZUE3DAG2o_j9n3K124GEPKqULZECVzgaTLUdk9z6ugCBB_ECW7Ve=s800, PlayEnabledGame=true, InstanceInstalled=true,
InstancePackageName=es.uan.eps.android.carolina.reversi, AchievementTotalCount=5, LeaderboardCount=0, RealTimeMultiplayerEnabled=true,
TurnBasedMultiplayerEnabled=true, AreSnapshotsEnabled=false, ThemeColor=0277bd, HasGamepadSupport=false},
MatchId=ChEKQ14uc_65BoQAhACGAAGARCDu6Kin-f548cB, CreatorId=p_1, CreationTimestamp=1466938835102, LastUpdaterId=p_1,
LastUpdatedTimestamp=1466940333014, PendingParticipantId=p_2, MatchStatus=1, TurnStatus=1, Description=Tu turno, Variant=-1, Data=[B@41d6afb8,
Version=7, Participants=[ParticipantEntity{ParticipantId=p_1, Player=PlayerEntity{PlayerId=g04118238548469996484, DisplayName=holaicaracola1,
HasDebugAccess=false, IconImageUrl=content://com.google.android.gms.games.background/images/d59ebf4d/69, IconImageUrl=http://
lh3.googleusercontent.com/r6xqj2d6bmtbPOY0nf0zy-tx2rAmGfQKnzpQYtAdmroWLoN_fjhvql-CjTad10yi-dFdg=s96, HiResImageUrl=content://
com.google.android.gms.games.background/images/d59ebf4d/70, HiResImageUrl=http://lh3.googleusercontent.com/r6xqj2d6bmtbPOY0nf0zy-
tx2rAmGfQKnzpQYtAdmroWLoN_fjhvql-CjTad10yi-dFdg=s282, RetrievedTimestamp=1466950213438, Title=Principiante,
LevelInfo=com.google.android.gms.games.PlayerLevelInfo@1e1b36, GamerTag=holaicaracola1, Name=Carolina Gutierrez,
BannerImageLandscapeUrl=content://com.google.android.gms.games.background/images/d59ebf4d/193, BannerImageLandscapeUrl=null,
BannerImagePortraitUrl=content://com.google.android.gms.games.background/images/d59ebf4d/194, BannerImagePortraitUrl=null}, Status=2,
ClientAddress=null, ConnectedToRoom=false, DisplayName=holaicaracola1, IconImage=content://com.google.android.gms.games.background/images/
d59ebf4d/69, IconImageUrl=http://lh3.googleusercontent.com/r6xqj2d6bmtbPOY0nf0zy-tx2rAmGfQKnzpQYtAdmroWLoN_fjhvql-CjTad10yi-dFdg=s96,
HiResImage=content://com.google.android.gms.games.background/images/d59ebf4d/70, HiResImageUrl=http://lh3.googleusercontent.com/
r6xqj2d6bmtbPOY0nf0zy-tx2rAmGfQKnzpQYtAdmroWLoN_fjhvql-CjTad10yi-dFdg=s282, Capabilities=0, Result=null}, ParticipantEntity
{ParticipantId=p_2, Player=PlayerEntity{PlayerId=g17175267141190348477, DisplayName=hola2caracola2, HasDebugAccess=false,
IconImage=content://com.google.android.gms.games.background/images/d59ebf4d/1, IconImageUrl=http://lh3.googleusercontent.com/
t0kFqc801aR_uP8J7JAqRLN2PAhouSKyWwS0RsIM4JI3Igf4bE3R31LaIux9sPpVr=s96, HiResImage=content://com.google.android.gms.background/
images/d59ebf4d/2, HiResImageUrl=http://lh3.googleusercontent.com/t0kFqc801aR_uP8J7JAqRLN2PAhouSKyWwS0RsIM4JI3Igf4bE3R31LaIux9sPpVr=s282,
RetrievedTimestamp=1467041127402, Title=Principiante, LevelInfo=com.google.android.gms.games.PlayerLevelInfo@1e1b36, GamerTag=hola2caracola2,
Name=TfgGpgs Carolina, BannerImageLandscapeUrl=content://com.google.android.gms.games.background/images/d59ebf4d/193,
BannerImageLandscapeUrl=null, BannerImagePortraitUrl=content://com.google.android.gms.games.background/images/d59ebf4d/194,
BannerImagePortraitUrl=null}, Status=2, CL
```

Figura 5.2: Ejemplo de mensaje del servidor

5.2. Compilación y pruebas

Para poder probar la aplicación es necesario firmar el fichero APK con la misma clave que se ha utilizado para generar el id de cliente OAuth 2.0 en la consola. En Android Studio se selecciona en el menú principal *Build* y luego *Generate Signed APK* y se muestra en cuadro como el de la figura 5.3.

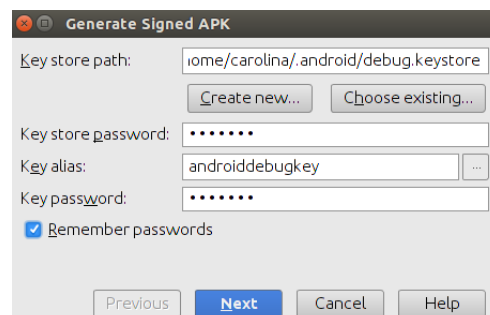


Figura 5.3: Firma APK en modo debug

Luego es posible instalar el APK firmado en el dispositivo físico para llevar a cabo las pruebas. Estando en la carpeta *platform-tools* del SDK se puede instalar la aplicación mediante el siguiente comando:

```
adb -d install -r '/ruta/al/fichero.apk'
```

Para ello dispositivo móvil debe tener habilitada la opción *USB debugging*.

6

Pruebas y Resultados

En este capítulo se detallan las pruebas realizadas y el resultado final obtenido: una aplicación publicada.

6.1. Pruebas

Para poder probar la aplicación ha sido necesario utilizar dos dispositivos físicos, cada uno registrado en la aplicación con una cuenta de correo electrónico de Gmail. Para ello ambas cuentas habían sido previamente autorizadas como testadores en la consola de desarrollador.

La aplicación se prueba mediante la instalación del fichero APK firmado tal y como se ha detallado en el capítulo anterior.

Tanto para verificar el correcto funcionamiento de la aplicación como para validarla se han llevado a cabo una serie de pruebas que han sido las siguientes:

- Pruebas de cada pantalla de manera individual.
- Pruebas de la correcta navegación entre pantallas.
- Pruebas de la repercusión de las acciones realizadas en una pantalla sobre otras pantallas.

En todos los casos se ha verificado que las pantallas mostradas son las que correspondían y que su funcionamiento es correcto.

6.2. Resultados

El resultado final de este Trabajo de Fin de Grado es una aplicación móvil para Android del juego multijugador Reversi publicado en la Play Store de Google. A continuación se muestran algunas imágenes de la aplicación final. En la figura 6.1 se muestra el menú principal de la aplicación.



Figura 6.1: Menú principal de la aplicación

En la figura 6.2 se muestra la pantalla principal del juego, el tablero.

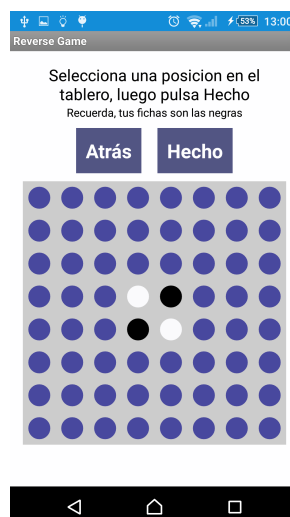


Figura 6.2: Pantalla de juego de la aplicación

En el anexo A se detalla cómo descargar la aplicación y un breve manual de usuario.

Además, en la consola de desarrollador se pueden obtener multitud de informes sobre la aplicación. Se puede extraer información sobre el número de descargas, la opinión de los usuarios o su valoración tal como se muestra en la figura 6.3.

PUNTUACIONES

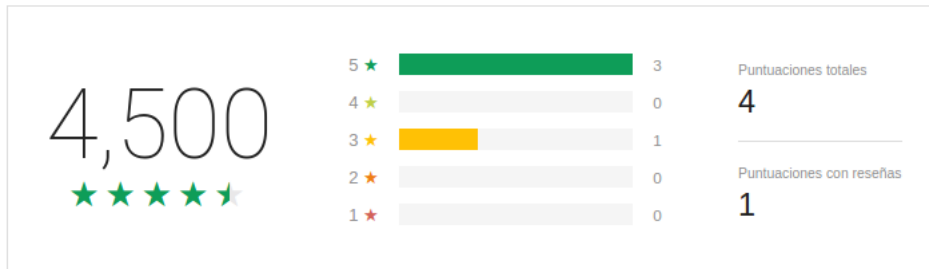


Figura 6.3: Valoración de la aplicación por los usuarios

7

Conclusiones y trabajo futuro

Este Trabajo de Fin de Grado ha consistido en el desarrollo de una aplicación Android, que utiliza el servicio de Google Play Games. Esta aplicación se ha centrado en el modo multijugador por turnos. Se ha implementado el juego Reversi, un juego para dos jugadores en el que el por turnos el objetivo es acabar con el mayor número de fichas sobre el tablero. Se ha conseguido de forma exitosa que dos usuarios puedan llevar a cabo una partida cada uno desde su dispositivo móvil.

Dado que este Trabajo de Fin de Grado era una prueba de concepto que trataba de ver qué era, en qué consistía y cómo se utilizaban los servicios de Google Play Games no se ha planteado un trabajo futuro. Es decir, no se han fijado posibles mejoras para llevar a cabo en la aplicación porque no era el objetivo del trabajo. Aunque sí podría plantearse la posibilidad de incluir como parte del contenido de la asignatura de Desarrollo de Aplicaciones para Dispositivos Móviles la utilización de servicios de Google Play Games pues este trabajo ha demostrado que simplifica significativamente el desarrollo de juegos y la interacción con servidores.

Durante el desarrollo de este proyecto, se han podido poner en práctica conocimientos adquiridos en diversas asignaturas del Grado en Ingeniería Informática tanto de las impartidas durante los primeros cursos como de las asignaturas de los últimos cursos. En *Análisis y Diseño de Software* y *Proyecto de Análisis y Diseño de Software* se aprendió a programar en Java, en las asignaturas de *Ingeniería del Software* y *Proyecto de Ingeniería del Software* se entendió cómo se debe gestionar un proyecto y cómo definir sus requisitos, y en la asignatura de *Desarrollo de Aplicaciones para Dispositivos Móviles* se aprendió a desarrollar aplicaciones móviles para Android. El haber podido emplear los conocimientos adquiridos a lo largo de la carrera ha hecho que la experiencia de la elaboración de este Trabajo de Fin de Grado haya sido enormemente enriquecedora.

Además, durante la implementación de este proyecto se ha aprendido a utilizar la consola de desarrollador de Google y las ventajas que presenta la utilización del servicio de Google Game Play. También se han asentado y mejorado los conocimientos sobre el desarrollo de aplicaciones Android. Por otra parte, se ha aprendido a utilizar varias herramientas como la consola de desarrollador y se ha mejorado en el uso de otras como Android Studio. Por lo que en general, el trabajo llevado a cabo ha sido más que satisfactorio.

Glosario

- Actividad** En programación Android se refiere a una componente de la aplicación que proporciona una pantalla que permite al usuario interactuar con ella con el fin de hacer algo. 21
- Android** Sistema operativo basado en el núcleo Linux. 1, 2, 5–11, 13, 15–17, 28, 31, 32
- API** *Application Programming Interface*, Conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizada por otro software como una capa de abstracción. 1, 6, 8, 21
- APK** Paquete que utiliza el sistema operativo Android para la distribución e instalación de aplicaciones móviles y middleware. 25–27
- Auto-emparejamiento** En el contexto de la aplicación se refiere a que el usuario se enfrenta a un jugador aleatorio. O bien se une a una partida ya existente o bien crea una y queda a la espera de que se añada un contrincante. 14, 23, 24
- Consola de desarrollador** Herramienta online proporcionada por Google que permite a los desarrolladores publicar y distribuir aplicaciones. 3, 7, 10, 16, 21, 23, 32
- Google Play Games Services** Servicio de juego online y SDK operado por Google para el sistema operativo Android. 2, 5, 6, 8, 13, 21
- Play Games** Centro de juego para los usuarios de Android que funciona como parte del servicio Google Play Games Services. 9
- Play Store** Plataforma de distribución digital de aplicaciones móviles para los dispositivos con sistema operativo Android. 1–3, 9, 10, 28
- SDK** *Software Development Kit*, Conjunto de herramientas de desarrollo de software que permite crear aplicaciones para un sistema concreto. 1, 7, 8, 21, 26

Referencias

- [1] *Google Play Games Services*. [Último acceso: 27/06/2016]. URL: <https://developers.google.com/games/services/>.
- [2] *Google Play Games Services - Achievements*. [Último acceso: 25/06/2016]. URL: <https://developers.google.com/games/services/common/concepts/achievements>.
- [3] *Google Play Games Services - Leaderboards*. [Último acceso: 25/06/2016]. URL: <https://developers.google.com/games/services/common/concepts/leaderboards>.
- [4] *Google Play Games Services - Realtime Multiplayer*. [Último acceso: 25/06/2016]. URL: <https://developers.google.com/games/services/common/concepts/realtimeMultiplayer>.
- [5] *Google Play Games Services - Turnbased Multiplayer*. [Último acceso: 25/06/2016]. URL: <https://developers.google.com/games/services/common/concepts/turnbasedMultiplayer>.
- [6] *Google Play Games Services - Quests*. [Último acceso: 25/06/2016]. URL: <https://developers.google.com/games/services/common/concepts/quests>.
- [7] *Google Play Games Services - Saved Games*. [Último acceso: 25/06/2016]. URL: <https://developers.google.com/games/services/common/concepts/savedgames>.
- [8] *Google Play Games Services - Player Stats*. [Último acceso: 25/06/2016]. URL: https://developers.google.com/games/services/android/stats#retrieving_player_stats_data.
- [9] *Google Play Games Services - Game Gifts*. [Último acceso: 25/06/2016]. URL: <https://developers.google.com/games/services/android/giftRequests>.
- [10] *Google Play Games Services - Nearby Connections*. [Último acceso: 25/06/2016]. URL: <https://developers.google.com/nearby/connections/overview>.
- [11] *Google Play Developer Console*. [Último acceso: 28/06/2016]. URL: <https://play.google.com/apps/publish/?hl=es>.
- [12] *Android Studio*. [Último acceso: 26/06/2016]. URL: <https://developer.android.com/studio/index.html>.
- [13] *Git*. [Último acceso: 26/06/2016]. URL: <https://git-scm.com>.
- [14] *Bitbucket*. [Último acceso: 26/06/2016]. URL: <https://bitbucket.org>.

- [15] *Libre Office*. [Último acceso: 26/06/2016]. URL: <https://es.libreoffice.org>.

Anexos



Manual de usuario

Este manual de usuario pretende ser una guía para la instalación y uso de la aplicación desarrollada que utiliza Google Play Games Services.

A.1. Instalación de la aplicación

Para poder descargar la aplicación Android desarrollada en este Trabajo de Fin de Grado que ha sido publicada en la Play Store se han de seguir los siguientes pasos:

1. Acceder a la Play Store de Google.
2. Buscar *ReverseGame*.
3. Seleccionar la opción *descargar*.

Nota: para poder usar la aplicación es necesario tener instalada la aplicación Play Games (Play Juegos en español) y tener actualizado Google Play services.

La aplicación desarrollada es la que se muestra como cuarto resultado en la figura A.1.

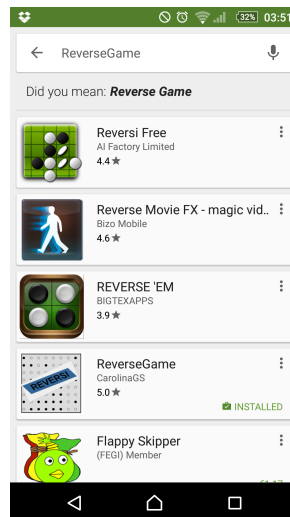


Figura A.1: Pantalla de la Play Store

A.2. Uso de la aplicación

Una vez descargada e iniciada la aplicación lo primero que se pedirá al usuario será que inicie sesión con su cuenta de correo de Gmail.

La primera vez que se acceda a la aplicación se le pedirá al usuario que cree un ID de usuario como se muestra en las figuras A.2 y A.3.

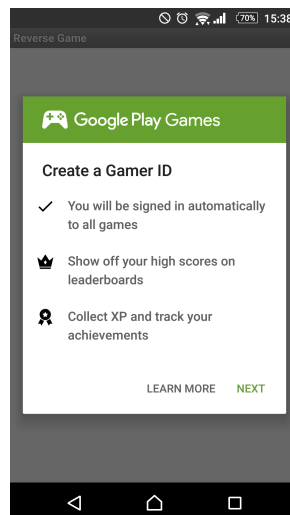


Figura A.2: Pantalla de inicio en la aplicación por primera vez

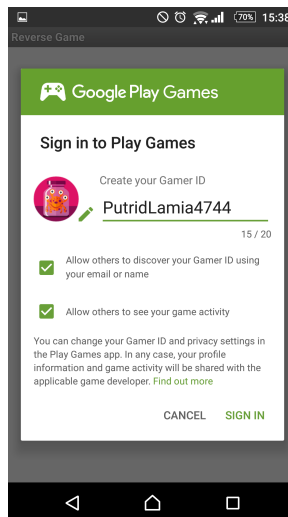


Figura A.3: Pantalla de configuración de los datos de usuario

A continuación, se le mostrará un menú de inicio como el de la figura A.4.



Figura A.4: Menú de inicio de la aplicación

Desde esta pantalla el usuario podrá:

- Iniciar una nueva partida:

Para ello el usuario debe seleccionar a su oponente (figura A.5). Podrá seleccionar un jugador automático, esto es, se enfrentará a un jugador aleatorio que haya seleccionado esta misma opción. O bien se creará una partida y quedará a la espera de que se una otro jugador o bien se unirá a una partida ya existente. También podrá jugar con jugadores cercanos, es decir, jugadores que se encuentren a su alrededor y tengan la opción de *Jugadores cerca* activa. Por último, podrá seleccionar enfrentarse a algún amigo o conocido que esté en sus círculos de Google+. En cualquier caso,

una vez que se haya seleccionado al oponente se le mostrará al jugador el tablero de juego donde debe indicar el movimiento que desea realizar (siempre empezará la partida el usuario que la crea). Tras efectuar el movimiento se le enviará de forma automática una invitación al jugador rival.

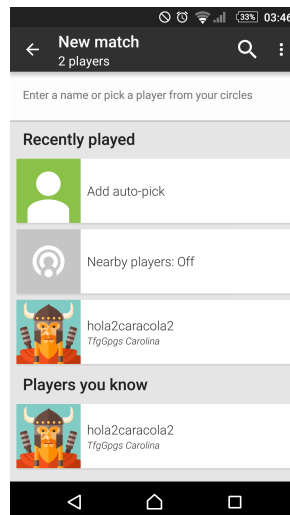


Figura A.5: Pantalla de selección de oponente

- Visualizar todas las partidas:

Se mostrarán invitaciones a partidas, aquellas partidas en las que sea el turno del usuario, aquellas en las que sea el turno del rival y las finalizadas. El usuario podrá aceptar invitaciones a nuevas partidas donde nada más unirse puede efectuar el movimiento. También podrá consultar las partidas en las que sea su turno y si lo desea, efectuar un movimiento. Por el contrario, el usuario no podrá consultar aquellas partidas en las que no sea su turno, deberá esperar a que mueva su oponente. En cuanto a las partidas finalizadas, el usuario podrá solicitar una revancha a su contrincante.

- Salir de la aplicación.



Manual del desarrollador

Este manual pretende ser una guía para el desarrollo de una aplicación Android que hace uso de los servicios de Google Play. Se detallan a lo largo de este anexo todos los pasos requeridos para la publicación del juego en la Play Store de Google, que incluyen el registro en la consola de desarrollador, la configuración del juego en la misma, y su posterior publicación.

B.1. Registro en la consola de desarrollador de Google Play

Para poder hacer uso de los servicios de Google Play es necesario estar en posesión de una cuenta de correo electrónico de Gmail y acceder a <https://play.google.com/apps/publish/signup/#>. Tras iniciar sesión en la consola de desarrollador será necesario aceptar el *Acuerdo de distribución para desarrolladores de Google Play* tal y como se muestra en la figura B.1.

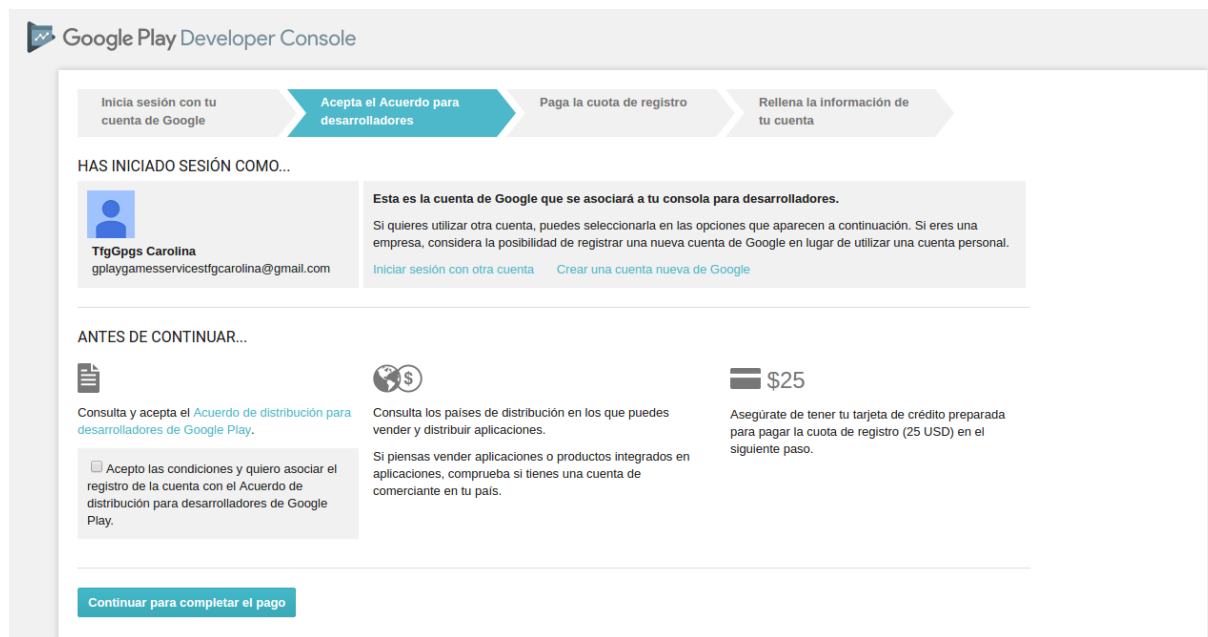


Figura B.1: Consola de desarrollador de Google Play

A continuación, se procede al pago de la cuota de registro por valor de 25\$. Por último, es necesario facilitar información de la cuenta.

B.2. Configuración del juego en la consola de desarrollador

Para llevar a cabo la configuración del juego en la consola de desarrollador se debe seleccionar la opción de *añadir juego nuevo* que aparece en la esquina superior derecha de la figura B.2.

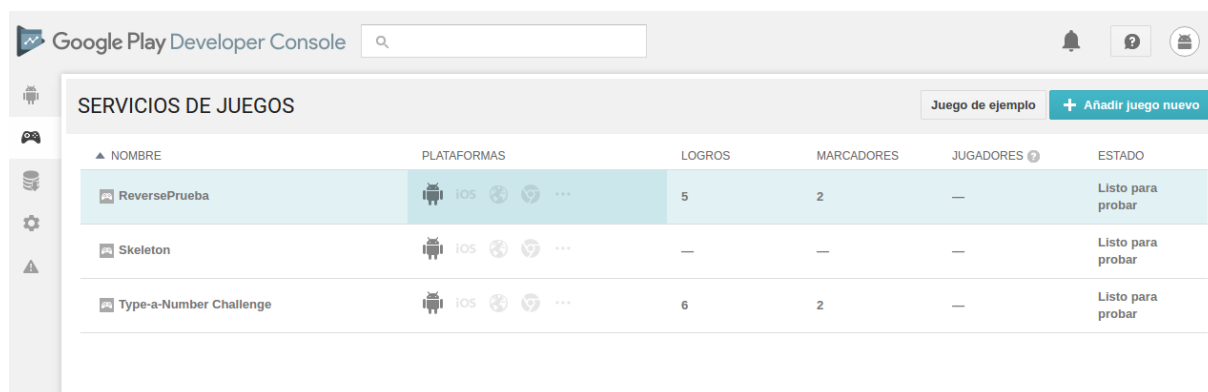


Figura B.2: Pantalla principal de la consola de desarrollador

A continuación, se pide indicar el nombre del juego, el tipo e indicar si se está haciendo

ya uso de las APIs de Google. Posteriormente, será necesario completar información del juego, vincular aplicaciones, crear los eventos, los logros y marcadores (tablas de clasificación) que tendrá el juego (además de incluir una imagen para cada uno de ellos), añadir las cuentas de correo electrónico de los usuarios autorizados para testear la aplicación y por último, una vez que todas las pantallas anteriores estén completas se permite la opción de publicar el juego (figura B.3).

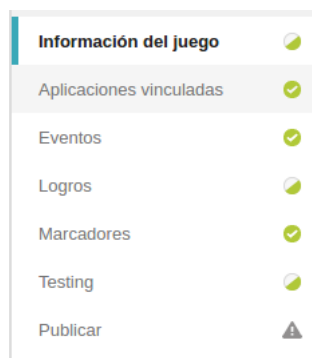


Figura B.3: Información requerida para la publicación del juego

B.2.1. Vincular aplicaciones

En la pantalla de aplicaciones vinculadas (figura B.4) se debe especificar el tipo de aplicación a desarrollar. Los pasos aquí detallados pueden variar dependiendo del tipo de aplicación seleccionada. En este caso, se explicarán los requeridos para vincular una aplicación para Android.

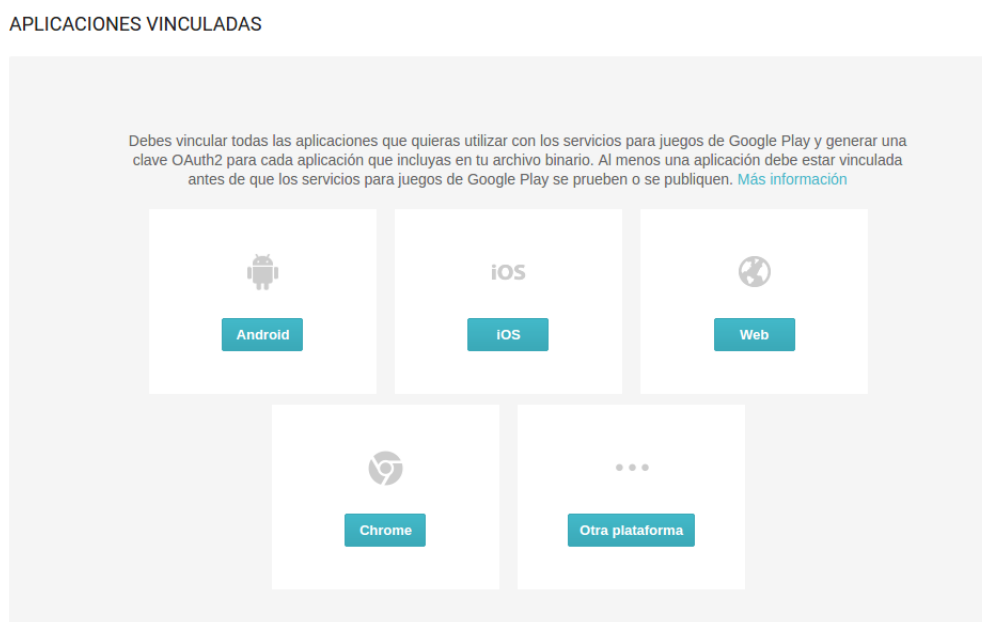


Figura B.4: Menú de selección para el tipo de aplicación

Una vez seleccionado el tipo de aplicación es necesario especificar el nombre del paquete tal y como esté en el proyecto de lo contrario no funcionará (figura B.5).

The screenshot shows the Google Play Console interface for linking an Android application. The main heading is 'VINCULAR UNA APLICACIÓN DE ANDROID' with a 'Guardar y continuar' button. The first step is 'PASO 1: INTRODUCE LA INFORMACIÓN SOBRE TU APLICACIÓN'. Under 'Idiomas (2)', 'Inglés (Estados Unidos) - en-US' is selected. The 'Nombre de la aplicación' field contains 'ReverseGameTFG' (14 de 30 caracteres). The 'Nombre del paquete' field contains 'es.uam.eps.android.carolina.reversi'. Below this is the 'AJUSTES DE MULTIJUGADOR' section with two toggle switches: 'Multijugador por turnos' (set to NO) and 'Multijugador en tiempo real' (set to SÍ). At the bottom is the 'ANTIPIRATERÍA' section with a toggle for 'Habilitar antipiratería' (set to SÍ). A left sidebar lists navigation options: Misiones, Información del juego, Aplicaciones vinculadas, Eventos, Logros, Marcadores, Testing, and Publicar.

Figura B.5: Vinculación una aplicación de Android

Para autenticar y autorizar la aplicación a utilizar los servicios de Google Play es necesario crear un cliente OAuth 2.0 de Android (figura B.6). Para obtener el ID de cliente se debe añadir la huella digital del certificado SHA1.

Normalmente, la *debug keystore* suele estar localizada en `/.android/debug.keystore` en el caso de sistemas operativos Mac o Linux y, en el caso de sistema operativo Windows, en `C:\Users\<USERNAME>\.android\debug.keystore`.

Por tanto, para obtener la información requerida en la consola hay que ejecutar en una terminal el siguiente comando:

```
keytool -exportcert -alias androiddebugkey -keystore /.android/debug.keystore -list -v
```

La contraseña solicitada es por defecto *android*. Una vez ejecutado el comando anterior copiamos la firma SHA1 obtenida. De este modo habremos obtenido el ID de la aplicación.

CREAR CLIENTE DE OAUTH DE ANDROID

Las solicitudes de la API se envían directamente a Google desde los dispositivos Android de los clientes. Google comprueba que cada solicitud procede de aplicación Android que coincida con el nombre de paquete y el de la huella digital del certificado de firma SHA1 que aparecen a continuación.

Nombre del paquete:	es.uam.eps.android.carolina.reversi
Huella digital de certificado de firma (SHA1):	<input type="text"/>
<div><input type="button" value="Confirmar"/> <input type="button" value="Cancelar"/></div>	

Figura B.6: Creación de cliente de OAuth

Luego, el ID de la aplicación hay que incluirlo en el proyecto, en el fichero *ids.xml* dentro de los recursos (figura B.7)



Figura B.7: Fichero ids.xml del proyecto

Además, en el caso de que el juego posea logros y marcadores también será necesario incluir sus ids en el fichero de recursos anteriormente mencionado.

B.3. Publicación del juego en la Play Store de Google

Una vez que se ha configurado el juego correctamente en la consola de desarrollador de Google y se ha implementado y probado el juego, éste se puede incorporar a la Play Store. Para ello se debe seleccionar *añadir nueva aplicación* tal como se muestra en la esquina superior de la figura B.8.



Figura B.8: Pantalla inicio de publicación del juego

A continuación, se debe añadir el fichero APK, cumplimentar información relativa al juego en la ficha de Play Store y responder a un cuestionario como el de la figura B.9.



Figura B.9: Clasificación de contenido de la publicación

Luego, una vez que se ha publicado el juego con éxito podemos añadir otras versiones subiendo de nuevo un fichero APK (figura B.10).

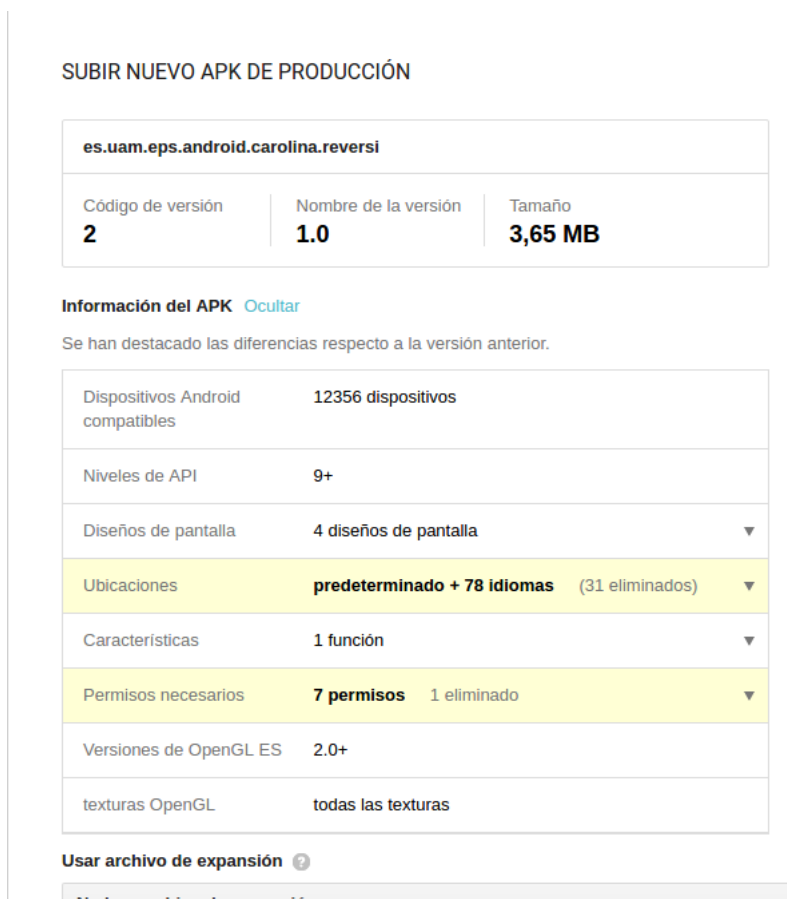


Figura B.10: Pantalla de éxito en la publicación

B.3.1. Firma del APK para la publicación

Una vez que el juego esté listo para ser publicado se debe subir a la consola de desarrollador el fichero APK firmado pero con una clave privada. Si no se posee una se puede crear. En Android Studio se selecciona en el menú principal *Build* y luego *Generate Signed APK* y se crea tal como se muestra en la figura B.11.

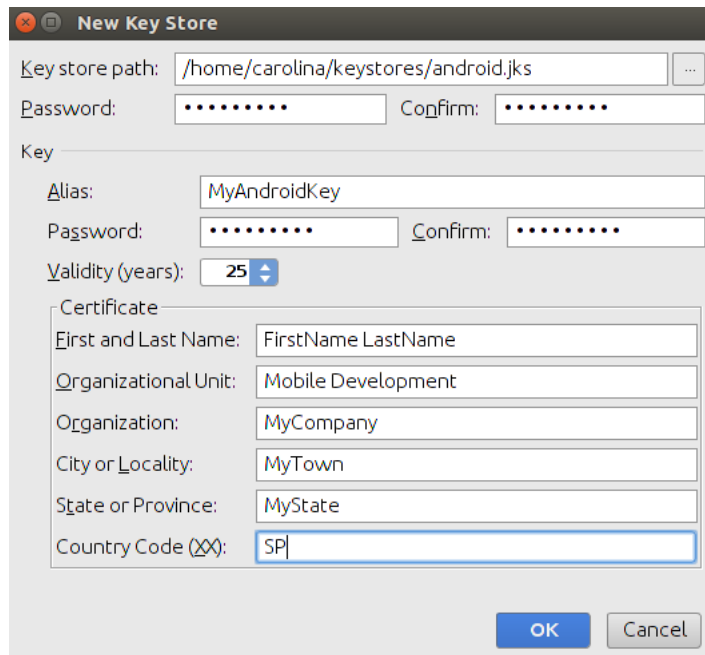


Figura B.11: Firma APK en modo release